Complementary Assignment

Aryaman Srivastava 2210110206

Q1.

Ans.

I explored POSIX shared memory primitives by examining two provided programs, namely shm-posix-producer-orig.c and shm-posix-consumer-orig.c.

The producer program initializes a shared memory segment, attaches it to its memory using mmap, and proceeds to write text into this segment. Conversely, the consumer program accesses the same shared memory segment, reads the text written by the producer, and then displays it on the screen.

To compile both programs, I used the gcc command with the -lrt flag: gcc -o prod shm-posix-producer-orig.c -lrt and gcc -o cons shm-posix-consumer-orig.c -lrt.

Upon executing the producer program (./prod), I observed the creation of the shared memory file under the directory /dev/shm/OS.

Subsequently, I ran the consumer program (./cons) to visualize the text previously written by the producer.

A screenshot of a computer

Description automatically generated

Q2.

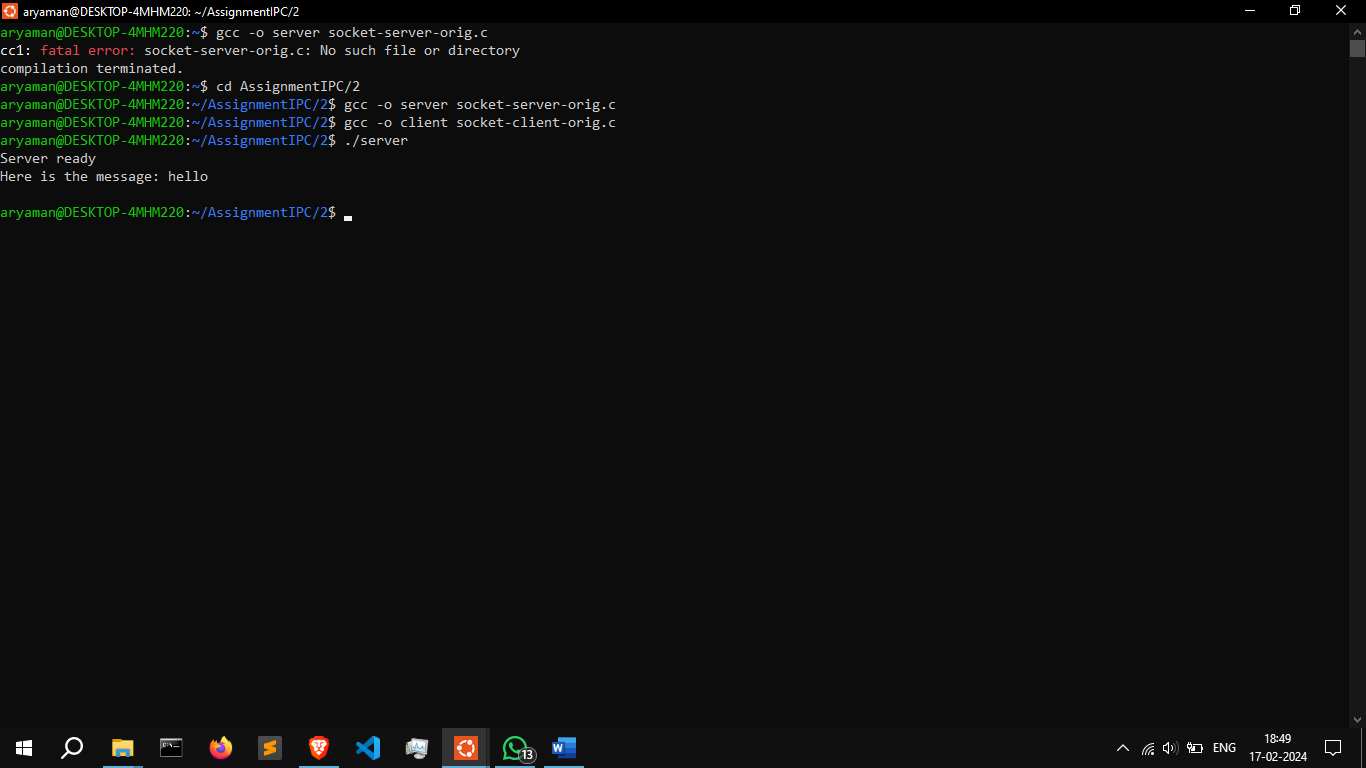
Ans. 2 Basic Socket Programming (Unix Domain Sockets):

Employed provided programs, socket-server-orig.c and socket-client-orig.c, to learn basic socket programming with Unix domain sockets.

Compiled both programs using gcc -o server socket-server-orig.c and gcc -o client socket-client-orig.c.

Started the server (./server) and initiated the client, sending a message to the server, which then displayed it.

Demonstrated bidirectional communication between the client and server using Unix domain sockets.



A screenshot of a computer

Description automatically generated

Q3.

Ans. Made two basic programs: one acts like a server (socket-server.c) and the other like a client (socket-client.c). They're like two computers talking to each other over the network.

In the client program, you type a message like "Hello from client," and it sends it to the server. The server then shows this message on its screen. Also, as soon as they connect, the server sends a message "Hello from server" to the client.

Made sure both programs compiled without errors, and when I ran them, I saw that they could talk to each other just fine.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Q4.

Ans.

**Pipe for communication between processes**

In one program, we send a basic message to the other program through the named pipe. The second program then receives this message and displays it on the screen.

After compiling both programs, I ran them to show how they can communicate with each other using named pipes. It's a way for different programs to share information with each other.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Q5.

Ans. A 4KB shared memory segment was filled with "free" strings to signify empty slots. The producer continuously produced strings, which were read by the consumer and then erased from shared memory.

To facilitate communication between the producer and consumer and manage slot allocation, a coordinated mechanism was implemented. This ensured synchronization between processes and proper utilization of shared memory slots.

The code was thoroughly tested to ensure correctness, verifying that shared memory slots were appropriately used and that processes were synchronized effectively.

A screenshot of a computer

Description automatically generated

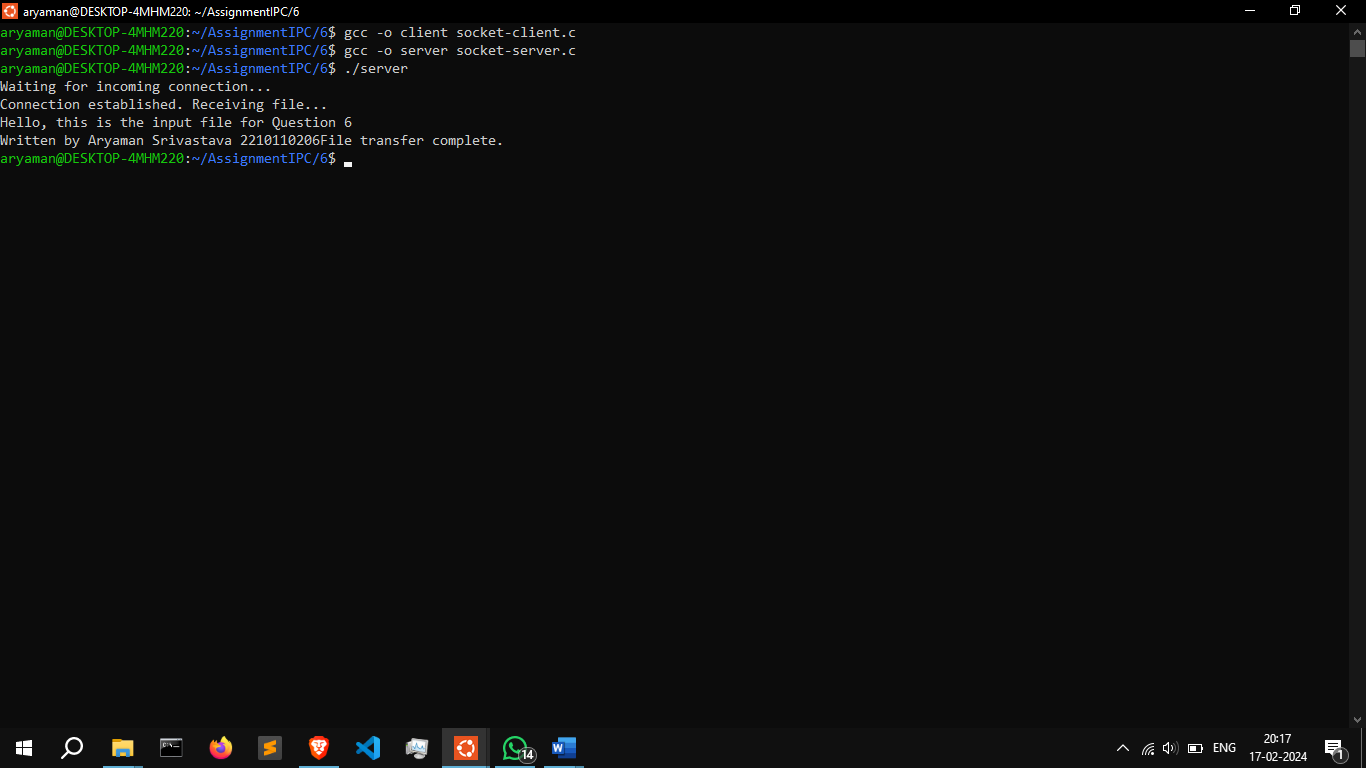
Q6.

Ans.

Two programs were developed: socket-client.c and socket-server.c, to enable file transfer using Unix domain sockets.

The client program took the filename as an argument, read the file in chunks, and sent the data over the socket to the server. Meanwhile, the server received the data from the client and displayed it on the screen.

To observe the file content transfer, the server and client were executed in separate terminal windows, allowing us to witness the file content transferred from the client being displayed on the server.



A screenshot of a computer

Description automatically generated